An Introduction to Sage

Arvind S Raj

Department of Cybersecurity Systems and Networks
Amrita University, India

1 February 2014 / FOSDEM

Outline

Bio

- Graduate CS student at Amrita University, India.
- Interested in computer security and Python.
- Sage: Component of graduate labs in Cryptography and used in CTF contests.

Introduction to Sage

- GPL licensed mathematics software.
- Unified interface to about 90 popular Python libraries.
- Two modes: command(like Python shell) and notebook(web interface).
- IPython shell and Python programming language.
- “sagerc“ file: $HOME/.sage/init.sage or $SAGE_STARTUP_FILE.

Arithmetic and built-in functions

- General arithmetic supported by an (I)Python shell.
  - ^ is exponent and ^^ is XOR.
  - For integers, / reduces to lowest fraction and // performs integer division.
- Support mathematical functions and constants with arbitrary precision.
  - `pi.n(digits=20)` = 3.14159265358979323855
  - `e.n(digits=25)` = 2.718281828459045235360287
  - `golden_ratio.n(prec=60)` = 1.6180339887498948
  - `n(sin(pi/3), prec=60)` = 0.86602540378443865
  - `sqrt(263).n(digits=20)` = 16.217274740226854774
  - `n(cos(5*pi/4), prec=60)` = -0.70710678118654752

Algebra

- Factorizing polynomials.
  - $factor(x^4 - 15x^3 + 84x^2 - 208x + 192)$ = $(x-3)(x-4)^3$
  - $factor(x^3 - 6x^2 + 11x - 6)$ = $(x-1)(x-2)(x-3)$
- Solving polynomial equations.
  - $solve([x^2 - 4x + 2 == -1], x)$ = $[x = 3, x = 1]$
  - Solutions to $x^2 + 3xy + y^2 = 0$ and $x - y = 4$ = $[[1.1055728, -2.8944272], [2.8944272, -1.1055728]]$
- Use find_root where solve does not work. Also useful to find solutions in a particular interval.
  - $solve(cos(t) == sin(t), t)$ = $[\sin(t) = \cos(t)]$
  - $find\_root(cos(t) == sin(t), 0, pi)$ = 0.785398163397

Number Theory

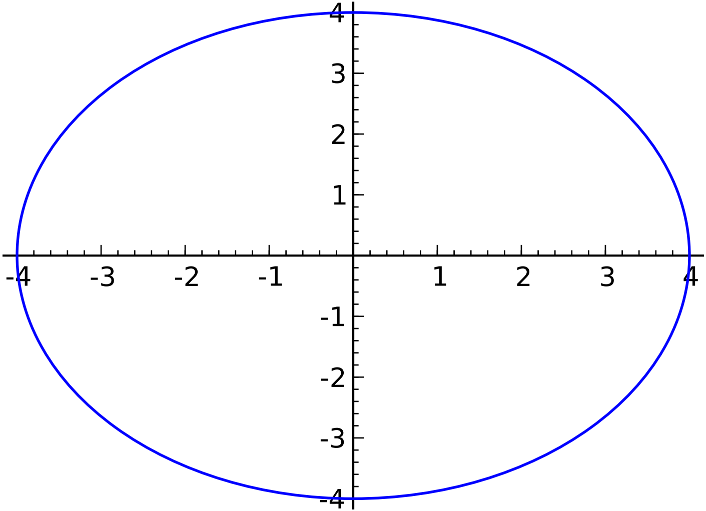
- Modulus: $mod(27, 12) = 3$ and $power\_mod(27, 2, 12) = 9$
- Primality test: $is\_prime(13) = \mathrm{True}$, $is\_prime(15) = \mathrm{False}$
- $prime\_range(1, 35) = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31]$.
  - Generator version: $primes(1, 35)$
- $primes\_first\_n(11) = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31]$
- $next\_prime(29) = 31$ and $previous\_prime = 23$
- $factorial(20) = 2432902008176640000$, $factor(20) = 2^2 \cdot 5$, $divisors(20) = [1, 2, 4, 5, 10, 20]$
- $gcd(10, 15) = 5$, $lcm(10, 15) = 30$

Calculus

- Differentiation
  - $diff(sin(x) + cos(x) = \cos(x) - \sin(x)$
  - $diff((sin(x\hat{}2)\hat{}3)) = 6\,x\cos\left(x^2\right)\sin\left(x^2\right)^2$
- Integration
  - $integral(cos(x) - sin(x)) = \cos(x) + \sin(x)$
  - $integral(6 * x * cos(x\hat{}2) * sin(x\hat{}2)\hat{}2, x) = \sin\left(x^2\right)^3$
- Partial differential and solving differential equations also possible!

Graph Plotting

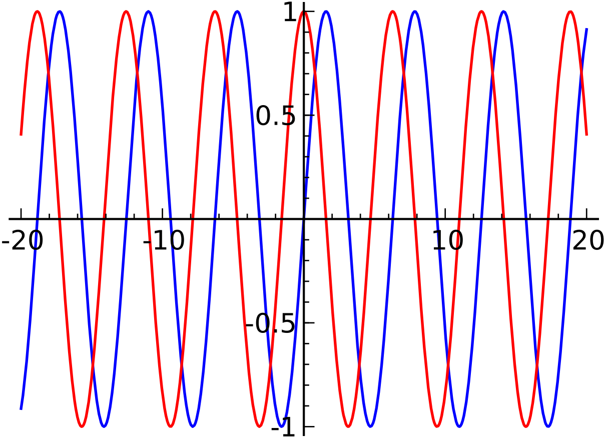
Circle of radius 4 centered at (0, 0): $c = circle((0,0),4)$



Graph Plotting(cont.)

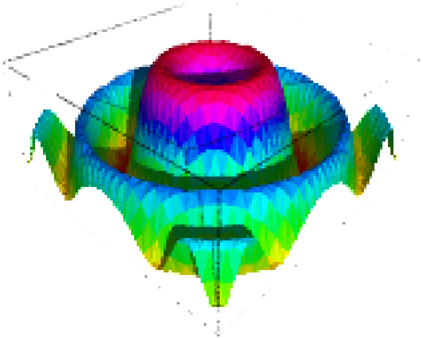
Multiple functions in same plot.

$$plot(sin(x), -20, 20, rgbcolor = (0, 0, 1)) + \\ plot(cos(x), -20, 20, rgbcolor = (1, 0, 0))$$



Graph Plotting(cont.)

$$f = \frac{sin(y*y+x*x)}{\sqrt{(x*x+y*y+.0001)}}: plot3d(f,(-3,3),(-3,3))$$



Matrix algebra

- Creating matrices: $m = Matrix([[1,2],[3,4],[5,6]])$
- Arithmetic operations
    - $P = Matrix([[1,2],[3,4]])$, $Q = Matrix([[7,8],[5,6]])$
    - P + Q = $\begin{pmatrix} 8 & 10 \\ 8 & 10 \end{pmatrix}$, P - Q = $\begin{pmatrix} -6 & -6 \\ -2 & -2 \end{pmatrix}$
    - P * Q = $\begin{pmatrix} 17 & 20 \\ 41 & 48 \end{pmatrix}$, 4 * P = $\begin{pmatrix} 4 & 8 \\ 12 & 16 \end{pmatrix}$
- P^3 = $\begin{pmatrix} 37 & 54 \\ 81 & 118 \end{pmatrix}$, P^{-1} = $\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$, |P| = -2
- More functions: is_singular, is_symmetric, is_skew_symmetric, is_invertible, is_square

LATEXand SageTEX

- LATEXrepresentation: latex(P)

  ```
  \left(\begin{array}{rr}
  1 & 2 \\
  3 & 4
  \end{array}\right)
  ```

- view(P): Display PDF(pdflatex)/HTML(MathJAX) depending on mode.
- SageTEX: Call Sage commands from LATEX.
  - Regular statement: \sage{pow_mod(27, 2, 12)}
  - Plots: \sageplot{plot(sin(x) + cos(x), -20, 20)}
  - \sageblock and \sagesilent: Embedding Sage code

Sage scripts

- Similar to Python scripts; .sage extension.
- import names from sage.all
- Run as sage <filename> <arguments> like Python.
- Other possibilities: profiling, compiling sage files(Cython), access C functions directly.

Other applications

- Interfacing with other algebra systems(GP/PARI, Singular, Maxima)
- Polynomials
- Combinatorics
- Graph and group theory
- Linear algebra
- Elliptic curves
- Advanced portions of everything discussed

References and further reading

- Sage tutorial: http://www.sagemath.org/doc/tutorial/index.html
- Thematic tutorials: http://www.sagemath.org/doc/thematic_tutorials/index.html
- Tutorials for those with some mathematics background: http://www.sagemath.org/doc/prep/index.html

Questions?

Thank you!